

Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Last Time Buy
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	17
Program Memory Size	4KB (4K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051t633-gmr

C8051T630/1/2/3/4/5

15. Interrupts	80
15.1. MCU Interrupt Sources and Vectors.....	81
15.1.1. Interrupt Priorities.....	81
15.1.2. Interrupt Latency	81
15.2. Interrupt Register Descriptions	82
15.3. INT0 and INT1 External Interrupts.....	87
16. EPROM Memory	89
16.1. Programming and Reading the EPROM Memory	89
16.1.1. EPROM Write Procedure	89
16.1.2. EPROM Read Procedure.....	90
16.2. Security Options	90
16.3. Program Memory CRC.....	91
16.3.1. Performing 32-bit CRCs on Full EPROM Content	91
16.3.2. Performing 16-bit CRCs on 256-Byte EPROM Blocks.....	91
17. Power Management Modes.....	92
17.1. Idle Mode.....	92
17.2. Stop Mode	93
17.3. Suspend Mode	93
18. Reset Sources	95
18.1. Power-On Reset.....	96
18.2. Power-Fail Reset/VDD Monitor	97
18.3. External Reset.....	98
18.4. Missing Clock Detector Reset	98
18.5. Comparator0 Reset	99
18.6. PCA Watchdog Timer Reset	99
18.7. EPROM Error Reset.....	99
18.8. Software Reset.....	99
19. Oscillators and Clock Selection	101
19.1. System Clock Selection.....	101
19.2. Programmable Internal High-Frequency (H-F) Oscillator	103
19.2.1. Internal Oscillator Suspend Mode.....	103
19.3. Programmable Internal Low-Frequency (L-F) Oscillator	105
19.3.1. Calibrating the Internal L-F Oscillator.....	105
19.4. External Oscillator Drive Circuit.....	106
19.4.1. External RC Example.....	108
19.4.2. External Capacitor Example.....	108
20. Port Input/Output	109
20.1. Port I/O Modes of Operation.....	110
20.1.1. Port Pins Configured for Analog I/O.....	110
20.1.2. Port Pins Configured For Digital I/O.....	110
20.1.3. Interfacing Port I/O to 5V Logic.....	111
20.2. Assigning Port I/O Pins to Analog and Digital Functions.....	112
20.2.1. Assigning Port I/O Pins to Analog Functions	112
20.2.2. Assigning Port I/O Pins to Digital Functions.....	112
20.2.3. Assigning Port I/O Pins to External Digital Event Capture Functions ...	113

C8051T630/1/2/3/4/5

24.1.1. Mode 0: 13-bit Counter/Timer	171
24.1.2. Mode 1: 16-bit Counter/Timer	172
24.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload.....	173
24.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only).....	174
24.2. Timer 2	179
24.2.1. 16-bit Timer with Auto-Reload.....	179
24.2.2. 8-bit Timers with Auto-Reload.....	180
24.2.3. Low-Frequency Oscillator (LFO) Capture Mode	181
24.3. Timer 3	185
24.3.1. 16-bit Timer with Auto-Reload.....	185
24.3.2. 8-bit Timers with Auto-Reload.....	186
24.3.3. Low-Frequency Oscillator (LFO) Capture Mode	187
25. Programmable Counter Array.....	191
25.1. PCA Counter/Timer	192
25.2. PCA0 Interrupt Sources.....	193
25.3. Capture/Compare Modules	194
25.3.1. Edge-triggered Capture Mode.....	195
25.3.2. Software Timer (Compare) Mode.....	196
25.3.3. High-Speed Output Mode	197
25.3.4. Frequency Output Mode	198
25.3.5. 8-bit, 9-bit, 10-bit and 11-bit Pulse Width Modulator Modes	198
25.3.5.1. 8-bit Pulse Width Modulator Mode.....	199
25.3.5.2. 9/10/11-bit Pulse Width Modulator Mode.....	200
25.3.6. 16-Bit Pulse Width Modulator Mode.....	201
25.4. Watchdog Timer Mode	202
25.4.1. Watchdog Timer Operation	202
25.4.2. Watchdog Timer Usage	203
25.5. Register Descriptions for PCA0.....	204
26. C2 Interface	210
26.1. C2 Interface Registers.....	210
26.2. C2 Pin Sharing	217
Document Change List.....	218
Contact Information.....	220

Table 5.13. Comparator Electrical CharacteristicsV_{DD} = 3.0 V, -40 to +85 °C unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Response Time: Mode 0, V _{cm} * = 1.5 V	CP0+ – CP0– = 100 mV	—	240	—	ns
	CP0+ – CP0– = -100 mV	—	240	—	ns
Response Time: Mode 1, V _{cm} * = 1.5 V	CP0+ – CP0– = 100 mV	—	400	—	ns
	CP0+ – CP0– = -100 mV	—	400	—	ns
Response Time: Mode 2, V _{cm} * = 1.5 V	CP0+ – CP0– = 100 mV	—	650	—	ns
	CP0+ – CP0– = -100 mV	—	1100	—	ns
Response Time: Mode 3, V _{cm} * = 1.5 V	CP0+ – CP0– = 100 mV	—	2000	—	ns
	CP0+ – CP0– = -100 mV	—	5500	—	ns
Common-Mode Rejection Ratio		—	1	4	mV/V
Positive Hysteresis 1	CP0HYP1-0 = 00	—	0	1	mV
Positive Hysteresis 2	CP0HYP1-0 = 01	2	5	8	mV
Positive Hysteresis 3	CP0HYP1-0 = 10	6	10	14	mV
Positive Hysteresis 4	CP0HYP1-0 = 11	12	20	28	mV
Negative Hysteresis 1	CP0HYN1-0 = 00	—	0	1	mV
Negative Hysteresis 2	CP0HYN1-0 = 01	2	5	8	mV
Negative Hysteresis 3	CP0HYN1-0 = 10	6	10	14	mV
Negative Hysteresis 4	CP0HYN1-0 = 11	12	20	28	mV
Inverting or Non-Inverting Input Voltage Range		-0.25	—	V _{DD} + 0.25	V
Input Offset Voltage		-7.5	—	7.5	mV
Power Specifications					
Power Supply Rejection		—	0.5	—	mV/V
Powerup Time		—	10	—	μs
Supply Current at DC	Mode 0	—	26	50	μA
	Mode 1	—	10	20	μA
	Mode 2	—	3	6	μA
	Mode 3	—	0.5	2	μA
Note: V _{cm} is the common-mode voltage on CP0+ and CP0–.					

6.3.2. Tracking Modes

The AD0TM bit in register ADC0CN enables "delayed conversions", and will delay the actual conversion start by three SAR clock cycles, during which time the ADC will continue to track the input. If AD0TM is left at logic 0, a conversion will begin immediately, without the extra tracking time. For internal start-of-conversion sources, the ADC will track anytime it is not performing a conversion. When the CNVSTR signal is used to initiate conversions, ADC0 will track either when AD0TM is logic 1, or when AD0TM is logic 0 and CNVSTR is held low. See Figure 6.2 for track and convert timing details. Delayed conversion mode is useful when AMUX settings are frequently changed, due to the settling time requirements described in Section "6.3.3. Settling Time Requirements" on page 36.

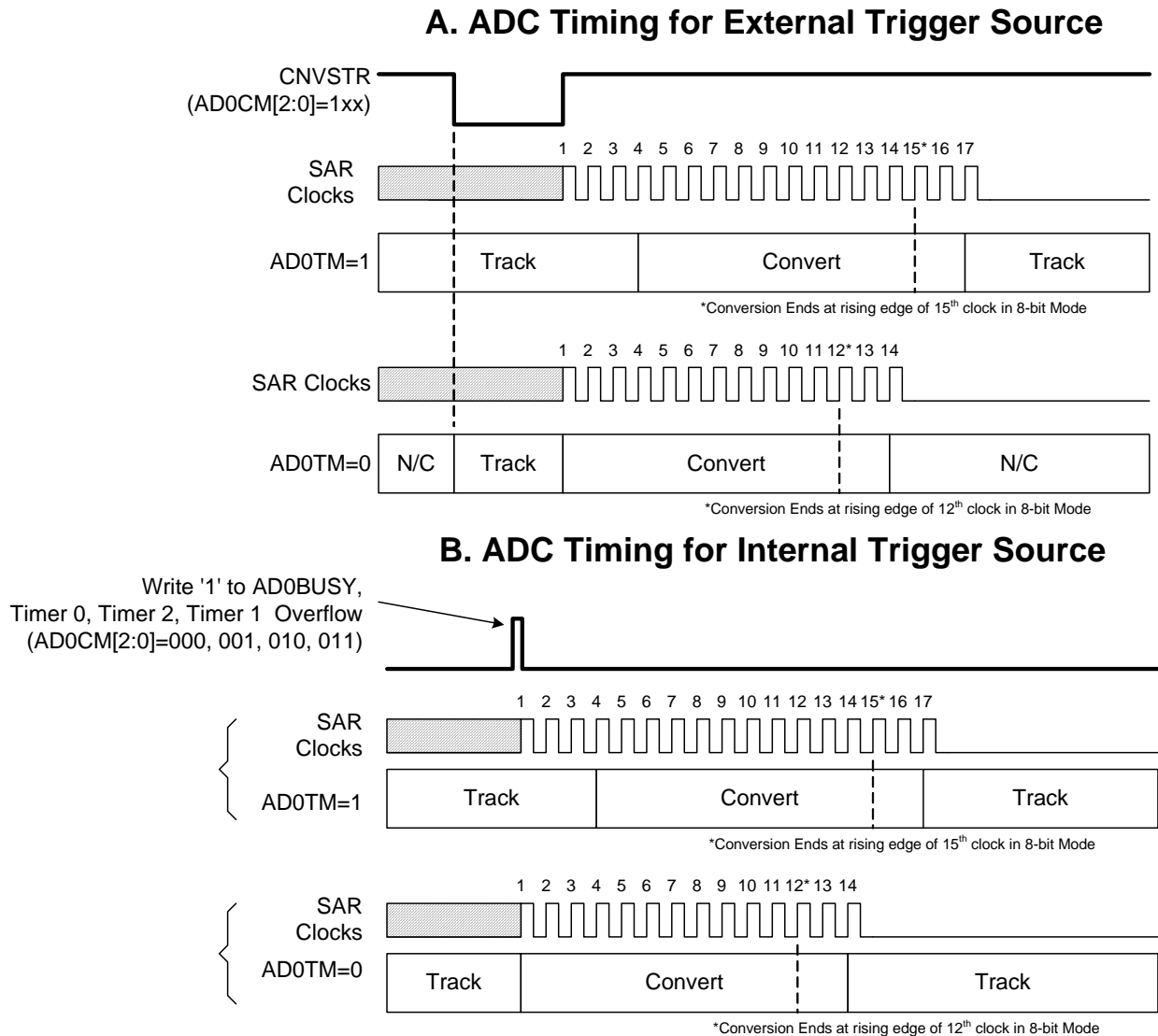


Figure 6.2. 10-Bit ADC Track and Conversion Example Timing

6.5. ADC0 Analog Multiplexer (C8051T630/2/4 only)

ADC0 on the C8051T630/2/4 uses an analog input multiplexer to select the positive input to the ADC. Any of the following may be selected as the positive input: Port 0 and 1 I/O pins, the on-chip temperature sensor, or the positive power supply (V_{DD}). The ADC0 input channel is selected in the AMX0P register described in SFR Definition 6.9.

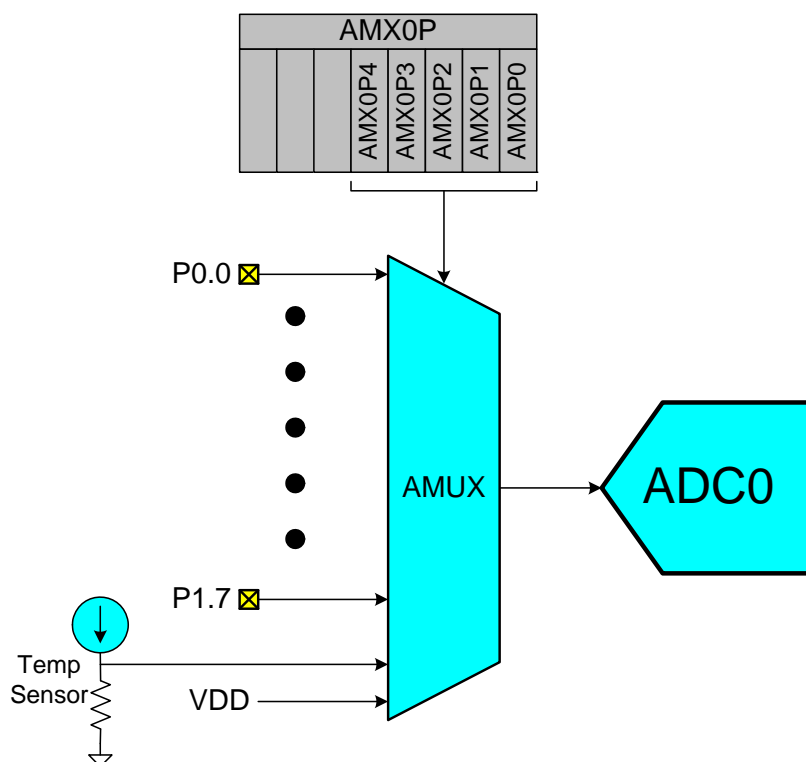


Figure 6.6. ADC0 Multiplexer Block Diagram

Important Note About ADC0 Input Configuration: Port pins selected as ADC0 inputs should be configured as analog inputs, and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to 0 the corresponding bit in register PnMDIN. To force the Crossbar to skip a Port pin, set to 1 the corresponding bit in register PnSKIP. See Section “20. Port Input/Output” on page 109 for more Port I/O configuration details.

12. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. The CIP-51 also includes on-chip debug hardware (see description in Section 26), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 12.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 25 MIPS Peak Throughput with 25 MHz Clock
- 0 to 25 MHz Clock Frequency
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

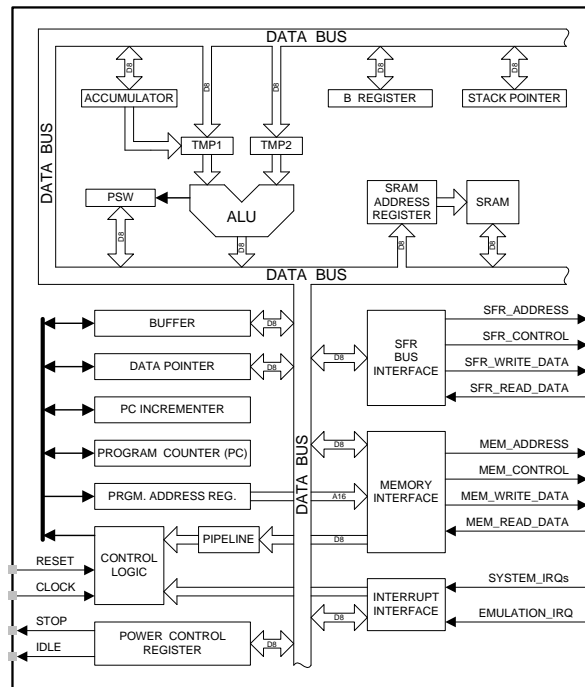


Figure 12.1. CIP-51 Block Diagram

14. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the C8051T630/1/2/3/4/5's resources and peripherals. The CIP-51 controller core duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the sub-systems unique to the C8051T630/1/2/3/4/5. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 14.1 lists the SFRs implemented in the C8051T630/1/2/3/4/5 device family.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, SCON0, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the data sheet, as indicated in Table 14.2, for a detailed description of each register.

Table 14.1. Special Function Register (SFR) Memory Map

F8	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	P0MAT	P0MASK	VDM0CN
F0	B	P0MDIN	P1MDIN				EIP1	PCA0PWM
E8	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	P1MAT	P1MASK	RSTSRC
E0	ACC	XBR0	XBR1	OSCLCN	IT01CF		EIE1	SMB0ADM
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2			
D0	PSW	REF0CN			P0SKIP	P1SKIP		SMB0ADR
C8	TMR2CN		TMR2RLL	TMR2RLH	TMR2L	TMR2H		
C0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	REG0CN
B8	IP	IDA0CN		AMX0P	ADC0CF	ADC0L	ADC0H	
B0		OSCXCN	OSCICN	OSCICL				
A8	IE	CLKSEL	EMI0CN					
A0	P2	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	
98	SCON0	SBUF0		CPT0CN		CPT0MD		CPT0MX
90	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H	IDA0L	IDA0H
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	
80	P0	SP	DPL	DPH		TOFFL	TOFFH	PCON
	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
	(bit addressable)							

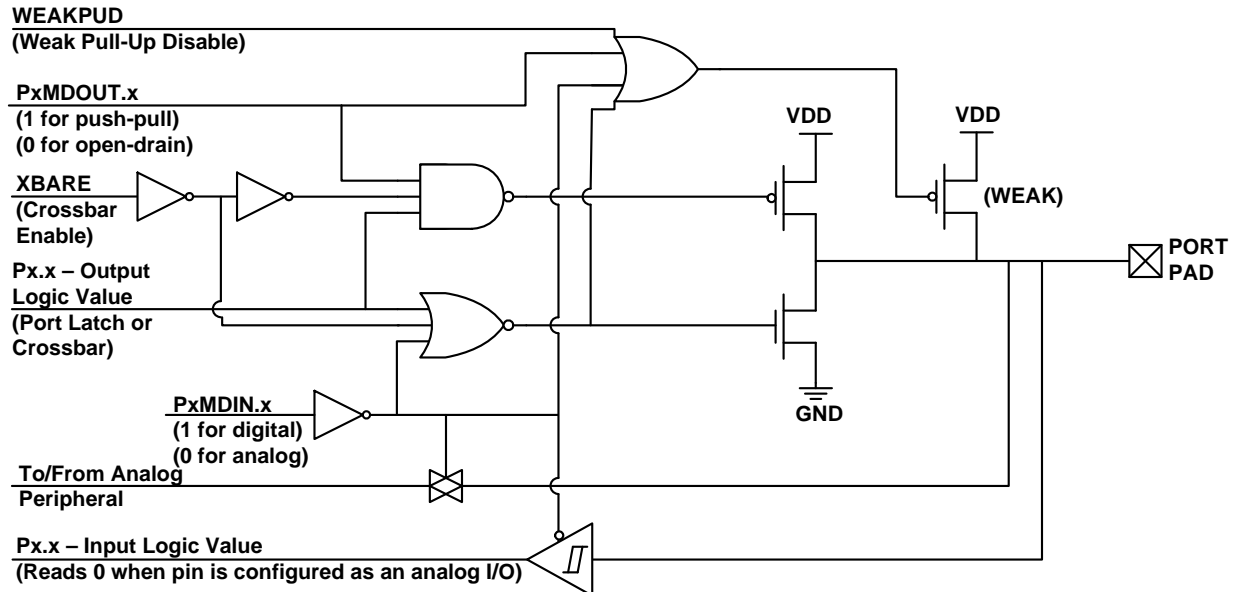


Figure 20.2. Port I/O Cell Block Diagram

20.1.3. Interfacing Port I/O to 5V Logic

All Port I/O configured for digital, open-drain operation are capable of interfacing to digital logic operating at a supply voltage higher than VDD and less than 5.25 V. An external pullup resistor to the higher supply voltage is typically required for most systems.

Important Note: In a multi-voltage interface, the external pullup resistor should be sized to allow a current of at least 150 μ A to flow into the Port pin when the supply voltage is between (VDD + 0.6 V) and (VDD + 1.0 V). Once the Port pin voltage increases beyond this range, the current flowing into the Port pin is minimal.

Table 21.1. SMBus Clock Source Selection

SMBCS1	SMBCS0	SMBus Clock Source
0	0	Timer 0 Overflow
0	1	Timer 1 Overflow
1	0	Timer 2 High Byte Overflow
1	1	Timer 2 Low Byte Overflow

The SMBCS1–0 bits select the SMBus clock source, which is used only when operating as a master or when the Free Timeout detection is enabled. When operating as a master, overflows from the selected source determine the absolute minimum SCL low and high times as defined in Equation 21.1. Note that the selected clock source may be shared by other peripherals so long as the timer is left running at all times. For example, Timer 1 overflows may generate the SMBus and UART baud rates simultaneously. Timer configuration is covered in Section “24. Timers” on page 169.

$$T_{HighMin} = T_{LowMin} = \frac{1}{f_{ClockSourceOverflow}}$$

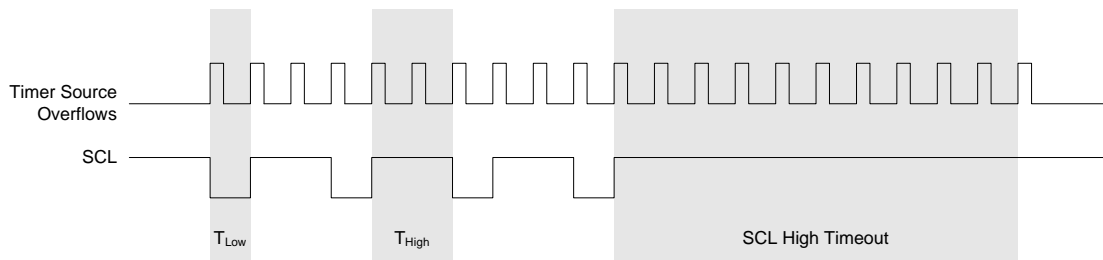
Equation 21.1. Minimum SCL High and Low Times

The selected clock source should be configured to establish the minimum SCL High and Low times as per Equation 21.1. When the interface is operating as a master (and SCL is not driven or extended by any other devices on the bus), the typical SMBus bit rate is approximated by Equation 21.2.

$$BitRate = \frac{f_{ClockSourceOverflow}}{3}$$

Equation 21.2. Typical SMBus Bit Rate

Figure 21.4 shows the typical SCL generation described by Equation 21.2. Notice that T_{HIGH} is typically twice as large as T_{LOW} . The actual SCL output may vary due to other devices on the bus (SCL may be extended low by slower slave devices, or driven low by contending master devices). The bit rate when operating as a master will never exceed the limits defined by equation Equation 21.1.

**Figure 21.4. Typical SMBus SCL Generation**

Setting the EXTHOLD bit extends the minimum setup and hold times for the SDA line. The minimum SDA setup time defines the absolute minimum time that SDA is stable before SCL transitions from low-to-high. The minimum SDA hold time defines the absolute minimum time that the current SDA value remains stable

C8051T630/1/2/3/4/5

21.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 21.2). The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a 1 to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a 1 to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 21.3 for more details.

Important Note About the SI Bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

21.4.2.1. Software ACK Generation

When the EHACK bit in register SMB0ADM is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

21.4.2.2. Hardware ACK Generation

When the EHACK bit in register SMB0ADM is set to 1, automatic slave address recognition and ACK generation is enabled. More detail about automatic slave address recognition can be found in Section 21.4.3. As a receiver, the value currently specified by the ACK bit will be automatically sent on the bus during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQ bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, further slave events will be ignored until the next START is detected, and no interrupt will be generated.

Table 21.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 21.5 for SMBus status decoding using the SMB0CN register.

of the slave address mask means that bit will be treated as a “don’t care” for comparison purposes. In this case, either a 1 or a 0 value are acceptable on the incoming slave address. Additionally, if the GC bit in register SMB0ADR is set to 1, hardware will recognize the General Call Address (0x00). Table 21.4 shows some example parameter settings and the slave addresses that will be recognized by hardware under those conditions.

Table 21.4. Hardware Address Recognition Examples (EHACK = 1)

Hardware Slave Address SLV[6:0]	Slave Address Mask SLVM[6:0]	GC bit	Slave Addresses Recognized by Hardware
0x34	0x7F	0	0x34
0x34	0x7F	1	0x34, 0x00 (General Call)
0x34	0x7E	0	0x34, 0x35
0x34	0x7E	1	0x34, 0x35, 0x00 (General Call)
0x70	0x73	0	0x70, 0x74, 0x78, 0x7C

SFR Definition 21.3. SMB0ADR: SMBus Slave Address

Bit	7	6	5	4	3	2	1	0
Name	SLV[6:0]							GC
Type	R/W							R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD7

Bit	Name	Function
7:1	SLV[6:0]	SMBus Hardware Slave Address. Defines the SMBus Slave Address(es) for automatic hardware acknowledgement. Only address bits which have a 1 in the corresponding bit position in SLVM[6:0] are checked against the incoming address. This allows multiple addresses to be recognized.
0	GC	General Call Address Enable. When hardware address recognition is enabled (EHACK = 1), this bit will determine whether the General Call Address (0x00) is also recognized by hardware. 0: General Call Address is ignored. 1: General Call Address is recognized.

Table 21.5. SMBus Status Decoding With Hardware ACK Generation Disabled (EHACK = 0)

Mode	Values Read				Current SMBus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X	1100
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X	1110
						Abort transfer.	0	1	X	—
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT.	0	0	X	1100
						End transfer with STOP.	0	1	X	—
						End transfer with STOP and start another transfer.	1	1	X	—
						Send repeated START.	1	0	X	1110
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT).	0	0	X	1000
Master Receiver	1000	1	0	X	A master data byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	1000
						Send NACK to indicate last byte, and send STOP.	0	1	0	—
						Send NACK to indicate last byte, and send STOP followed by START.	1	1	0	1110
						Send ACK followed by repeated START.	1	0	1	1110
						Send NACK to indicate last byte, and send repeated START.	1	0	0	1110
						Send ACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	1	1100
						Send NACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	0	1100

22. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in Section “22.1. Enhanced Baud Rate Generation” on page 149). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

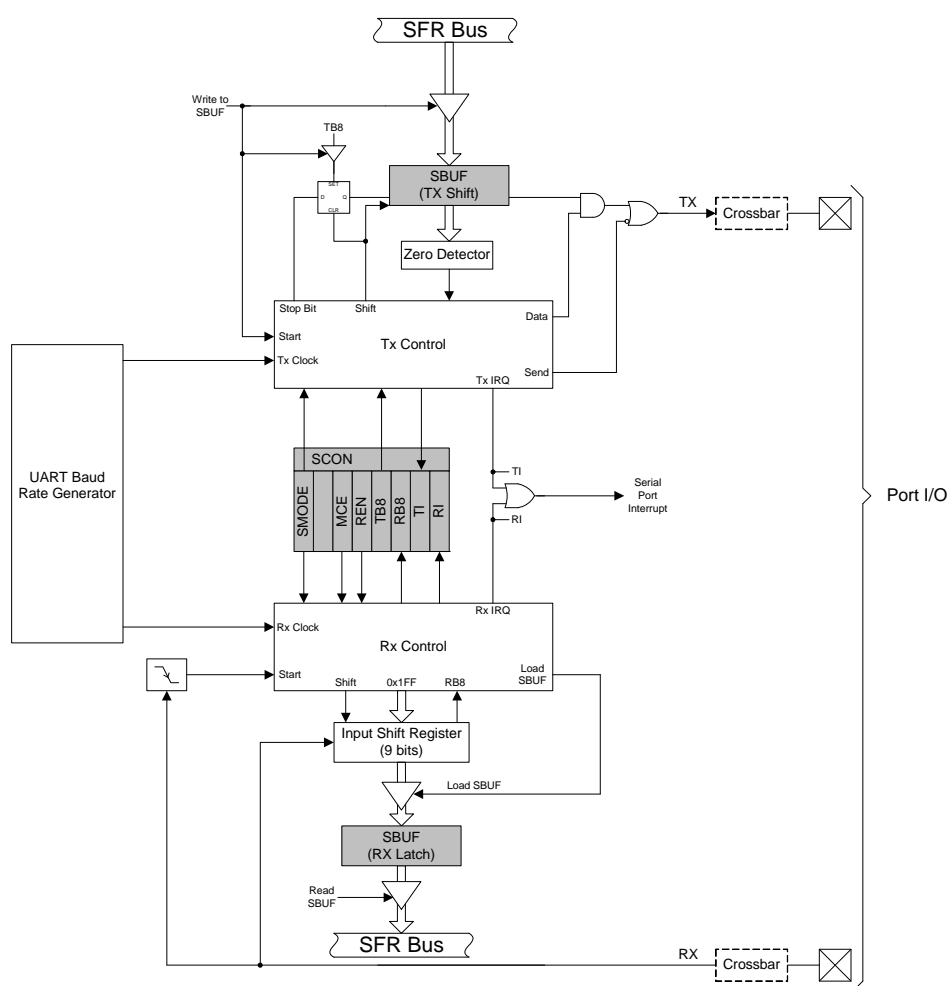


Figure 22.1. UART0 Block Diagram

SFR Definition 23.2. SPI0CN: SPI0 Control

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	MODF	RXOVRN	NSSMD[1:0]		TXBMT	SPIEN
Type	R/W	R/W	R/W	R/W	R/W		R	R/W
Reset	0	0	0	0	0	1	1	0

SFR Address = 0xF8; Bit-Addressable

Bit	Name	Function
7	SPIF	SPI0 Interrupt Flag. This bit is set to logic 1 by hardware at the end of a data transfer. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
6	WCOL	Write Collision Flag. This bit is set to logic 1 if a write to SPI0DAT is attempted when TXBMT is 0. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
5	MODF	Mode Fault Flag. This bit is set to logic 1 by hardware when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
4	RXOVRN	Receive Overrun Flag (valid in slave mode only). This bit is set to logic 1 by hardware when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
3:2	NSSMD[1:0]	Slave Select Mode. Selects between the following NSS operation modes: (See Section 23.2 and Section 23.3). 00: 3-Wire Slave or 3-Wire Master Mode. NSS signal is not routed to a port pin. 01: 4-Wire Slave or Multi-Master Mode (Default). NSS is an input to the device. 1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.
1	TXBMT	Transmit Buffer Empty. This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.
0	SPIEN	SPI0 Enable. 0: SPI disabled. 1: SPI enabled.

24. Timers

Each MCU includes four counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timer for use with the ADC, SMBus, or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 and Timer 3 offer 16-bit and split 8-bit timer functionality with auto-reload. Additionally, Timer 3 offers the ability to be clocked from the external oscillator while the device is in Suspend mode, and can be used as a wake-up source. This allows for implementation of a very low-power system, including RTC capability.

Timer 0 and Timer 1 Modes:	Timer 2 Modes:	Timer 3 Modes:
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer		
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)		

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 24.1 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 and Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

C8051T630/1/2/3/4/5

24.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 24.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bit (T2XCLK in TMR2CN), as follows:

T2MH	T2XCLK	TMR2H Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

T2ML	T2XCLK	TMR2L Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

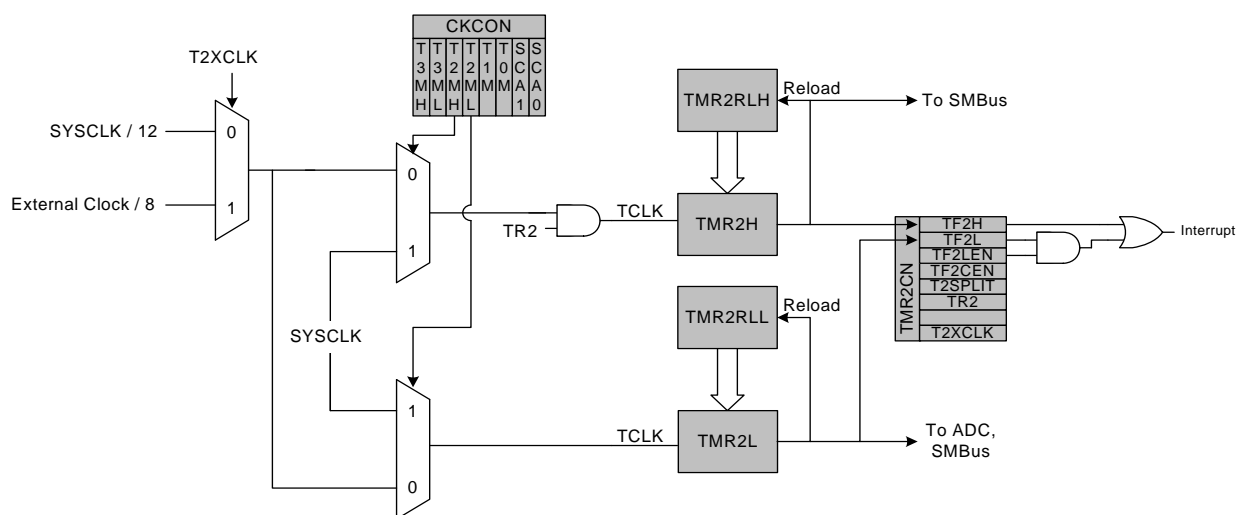


Figure 24.5. Timer 2 8-Bit Mode Block Diagram

C8051T630/1/2/3/4/5

SFR Definition 24.8. TMR2CN: Timer 2 Control

Bit	7	6	5	4	3	2	1	0
Name	TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2		T2XCLK
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC8; Bit-Addressable

Bit	Name	Function
7	TF2H	Timer 2 High Byte Overflow Flag. Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF2L	Timer 2 Low Byte Overflow Flag. Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.
5	TF2LEN	Timer 2 Low Byte Interrupt Enable. When set to 1, this bit enables Timer 2 Low Byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
4	TF2CEN	Timer 2 Low-Frequency Oscillator Capture Enable. When set to 1, this bit enables Timer 2 Low-Frequency Oscillator Capture Mode. If TF2CEN is set and Timer 2 interrupts are enabled, an interrupt will be generated on a falling edge of the low-frequency oscillator output, and the current 16-bit timer value in TMR2H:TMR2L will be copied to TMR2RLH:TMR2RLL.
3	T2SPLIT	Timer 2 Split Mode Enable. When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload. 0: Timer 2 operates in 16-bit auto-reload mode. 1: Timer 2 operates as two 8-bit auto-reload timers.
2	TR2	Timer 2 Run Control. Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode.
1	Unused	Unused. Read = 0b; Write = Don't Care
0	T2XCLK	Timer 2 External Clock Select. This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 2 clock is the system clock divided by 12. 1: Timer 2 clock is the external clock divided by 8 (synchronized with SYSCLK).

C8051T630/1/2/3/4/5

C2 Register Definition 26.2. DEVICEID: C2 Device ID

Bit	7	6	5	4	3	2	1	0
Name	DEVICEID[7:0]							
Type	R/W							
Reset	0	0	0	1	0	1	1	1

C2 Address: 0x00

Bit	Name	Function
7:0	DEVICEID[7:0]	Device ID. This read-only register returns the 8-bit device ID: 0x17 (C8051T630/1/2/3/4/5).

C2 Register Definition 26.3. REVID: C2 Revision ID

Bit	7	6	5	4	3	2	1	0
Name	REVID[7:0]							
Type	R/W							
Reset	Varies	Varies	Varies	Varies	Varies	Varies	Varies	Varies

C2 Address: 0x01

Bit	Name	Function
7:0	REVID[7:0]	Revision ID. This read-only register returns the 8-bit revision ID. For example: 0x00 = Revision A.

C2 Register Definition 26.12. CRC2: CRC Byte 2

Bit	7	6	5	4	3	2	1	0
Name	CRC[23:16]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

C2 Address: 0xAB

Bit	Name	Function
7:0	CRC[23:16]	CRC Byte 2. See Section “16.3. Program Memory CRC” on page 91.

C2 Register Definition 26.13. CRC3: CRC Byte 3

Bit	7	6	5	4	3	2	1	0
Name	CRC[31:24]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

C2 Address: 0xAC

Bit	Name	Function
7:0	CRC[31:24]	CRC Byte 3. See Section “16.3. Program Memory CRC” on page 91.

DOCUMENT CHANGE LIST

Revision 0.2 to Revision 1.0

- Updated electrical specification tables based on test, characterization, and qualification data.
- Updated figures and text to correct minor typographical errors throughout document.
- Updated package definitions to include all possible vendor information, and JEDEC-standard drawings.